Grocery Web Application

Version: 1.0
Date: September 6, 2025
Project ID: PNT2025TMID14542
Team: Umair Borkar (Lead), Abhishek Pawar, Utkarsha Desai, Snehal Mali

1. Introduction

1.1. Project Overview

The Grocery Web Application is a comprehensive, full-stack Grocery Management platform designed to revolutionize the online grocery shopping experience. It bridges the gap between traditional retail and modern digital convenience by offering a fast, intuitive, and secure interface for a diverse user base, from tech-savvy shoppers to homemakers managing daily essentials.

Developed as part of the SmartBridge Externship program, this project embodies the application of cutting-edge web technologies and agile collaborative practices to solve a real-world need.

1.2. Problem Statement

The modern consumer demands efficiency, variety, and reliability in online shopping. Existing solutions can often be complex, slow, or lack robust management tools for sellers. This application aims to solve these challenges by providing a user-centric platform that is equally powerful for end-users shopping for groceries and for administrators and sellers managing the digital storefront.

1.3. Team Details

- **Team ID:** PNT2025TMID14542
- **Team Size:** 4 Members
- **Team Lead:** Umair Borkar
- **Team Members:** Abhishek Pawar, Utkarsha Desai, Snehal Mali

2. Overall Description

2.1. Product Perspective

This is a standalone web application that operates on a client-server model. The frontend, built with Angular, handles all user interactions, while the backend, powered by Node.js and Express, manages business logic, data persistence, and security. MongoDB serves as the primary database for its flexibility and scalability in handling e-commerce data.

2.2. User Classes and Characteristics

1. **Customers (Users):** The primary end-users. They browse categories, view products, manage a cart, and place orders. They require an intuitive, fast, and trustworthy interface.

2. **Administrators (Admins):** Responsible for the overall health of the platform. They manage users,

view transactions, handle customer feedback, and ensure system stability.

3. **Sellers:** A subset of privileged users (can be merged with Admin for a simpler v1). They manage product listings, update inventory stock, and process incoming orders.

2.3. Operating Environment

- **Frontend:** Angular application served via a web server (e.g., Nginx, Apache) or directly through Angular CLI in development. Compatible with all modern browsers (Chrome, Firefox, Safari, Edge).
- **Backend:** Node.js runtime environment hosted on a cloud server (e.g., AWS, Azure, Heroku) or a local server.
- **Database:** MongoDB, hosted via MongoDB Atlas (cloud) or on a local instance.
- **Development:** Visual Studio Code, Git for version control, and Node Package Manager (npm).

3. System Features and Requirements

3.1. User Side Features

- **User Registration & Authentication:** Secure user sign-up and login system.
- **Product Browsing:** Intuitive interface to explore products organized by categories.
- **Product Search & Filter:** Ability to search for products and filter results.
- **Shopping Cart:** Virtual cart to add, update, or remove items before purchase.
- **Checkout Process:** A secure, multi-step process to enter shipping details, select payment method, and place orders.
- **Order History:** A user profile section to view past orders and track their status.

3.2. Admin/Seller Side Features

- **Admin Dashboard:** A centralized panel to view platform analytics (sales, orders, users).
- **Product Management:** Full CRUD (Create, Read, Update, Delete) operations on products and categories.
- **Order Management:** View, update status (e.g., Processing, Shipped, Delivered), and manage all customer orders.
- **User Feedback & Support:** Interface to monitor and respond to customer inquiries and feedback.
- **Inventory Tracking:** Tools to monitor and update product stock levels.

3.3. Non-Functional Requirements

- **Performance:** Pages should load in under 3 seconds. The system should handle a growing number of products and users.

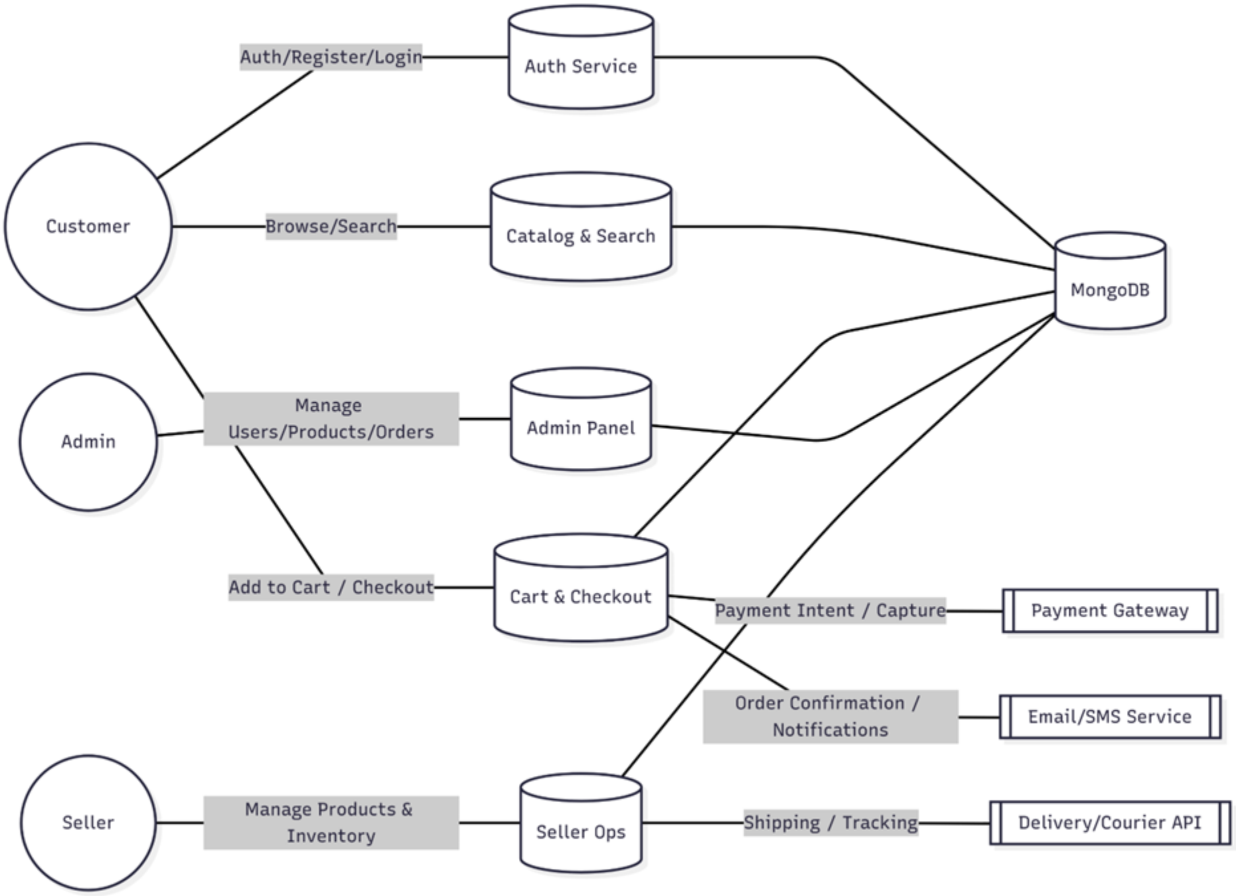
- **Security:** All user data and passwords must be encrypted. The application must be protected against common web vulnerabilities (XSS, SQL Injection). Payment information must be processed securely via a certified gateway.
- **Usability:** The interface must be responsive and provide an excellent experience on both desktop and mobile devices.
- **Reliability:** The system should achieve 99.9% uptime, with robust error handling and logging.

4. Design & Architecture

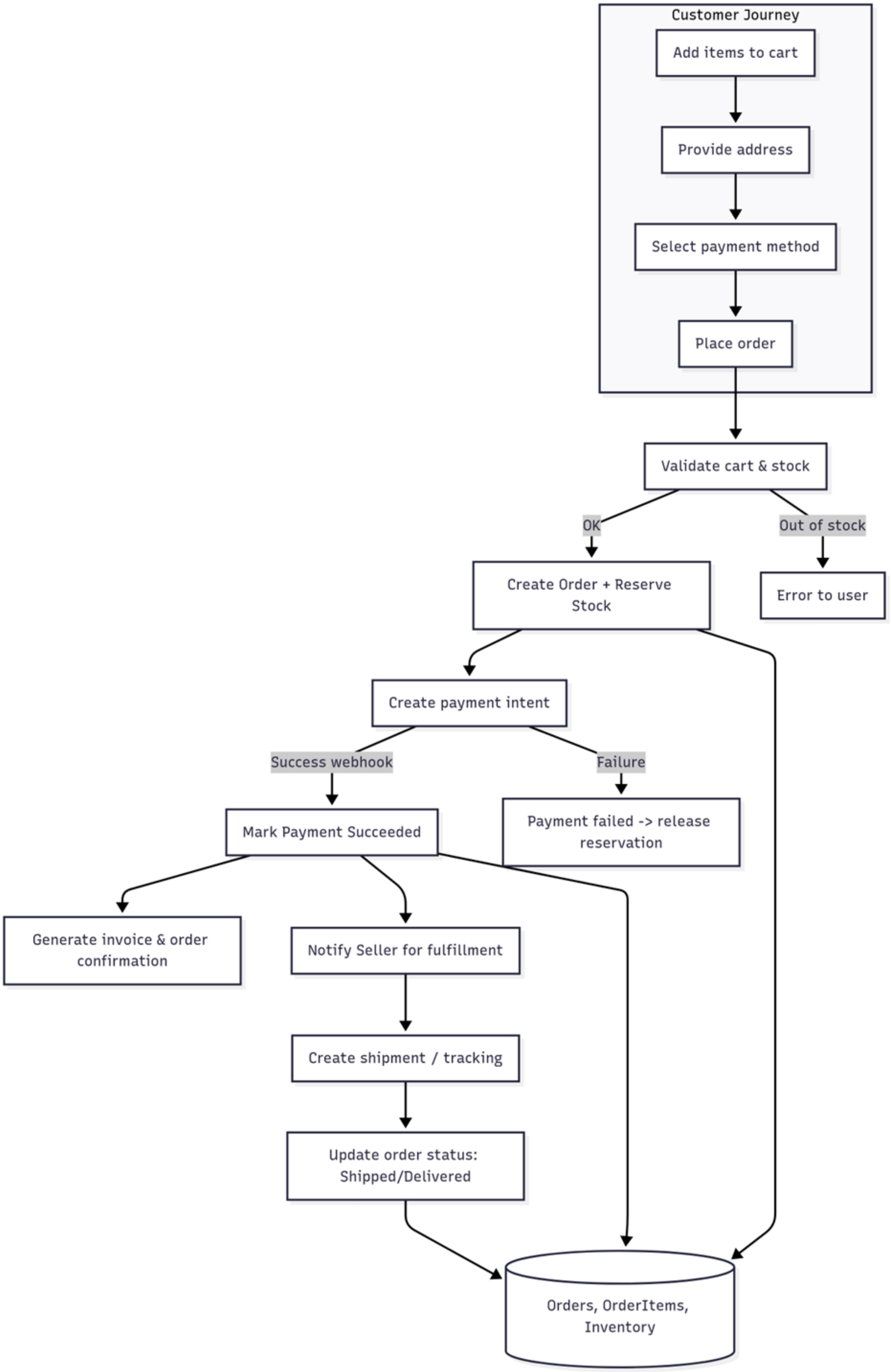
4.1. Technical Architecture

The application follows a modular 3-tier architecture for separation of concerns, scalability, and maintainability.

Level 0:



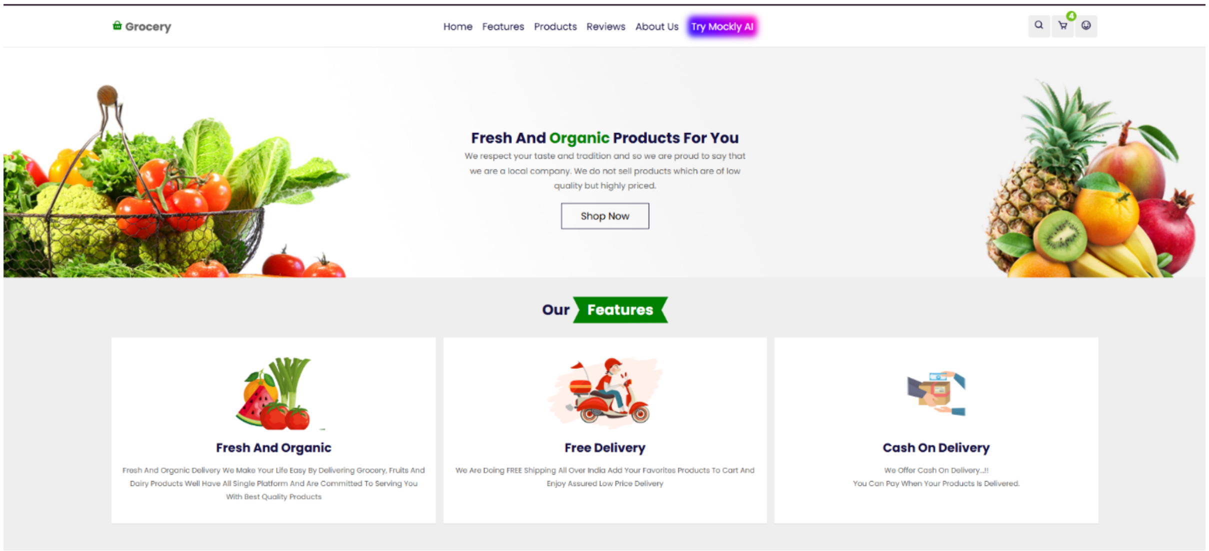
Level 1:


Customer Journey
Add items to cart
Provide address
Select payment method
Place order
Validate cart & stock
OK
Out of stock
Create Order + Reserve Stock
Error to user
Create payment intent
Success webhook
Failure
Mark Payment Succeeded
Payment failed -> release reservation
Generate invoice & order confirmation
Notify Seller for fulfillment
Create shipment / tracking
Update order status: Shipped/Delivered
Orders, OrderItems, Inventory

Tiers:

1. **Presentation Tier (Frontend - Angular):** Renders the user interface and captures user input.
2. **Application Tier (Backend - Node.js/Express):** Contains the business logic, processes requests, and communicates with the database and external services. It is structured as a set of microservices:

    ○ User Service



Fresh And Organic Products For You

We respect your taste and tradition and so we are proud to say that we are a local company. We do not sell products which are of low quality but highly priced.

Our Features

Fresh And Organic

Fresh And Organic Delivery We Make Your Life Easy By Delivering Grocery, Fruits And Dairy Products Well Have All Single Platform And Are Committed To Serving You With Best Quality Products

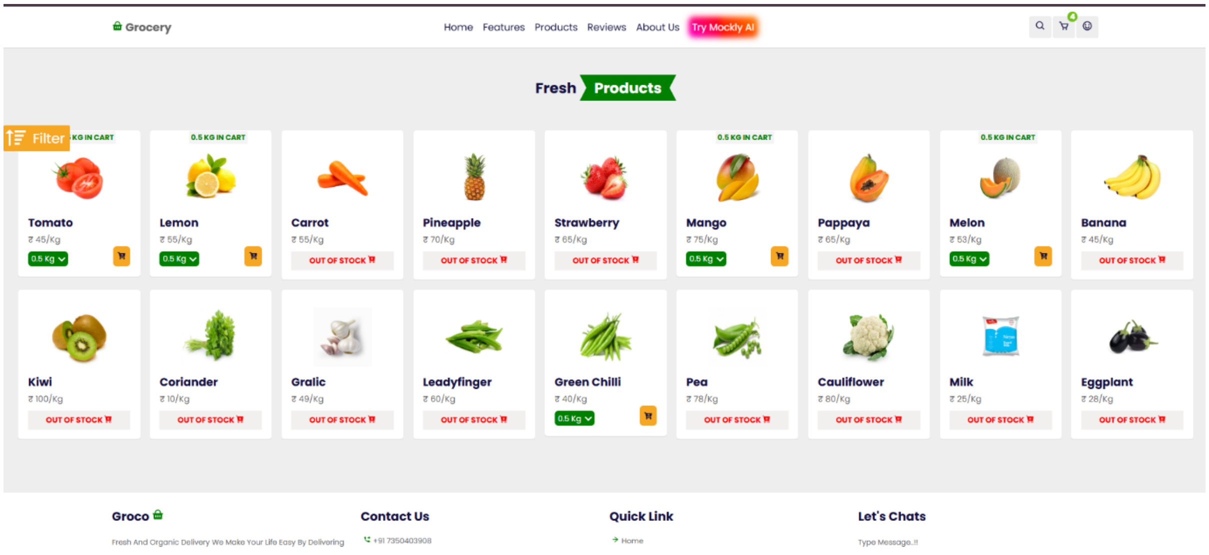
Free Delivery

We Are Doing FREE Shipping All Over India Add Your Favorites Products To Cart And Enjoy Assured Low Price Delivery

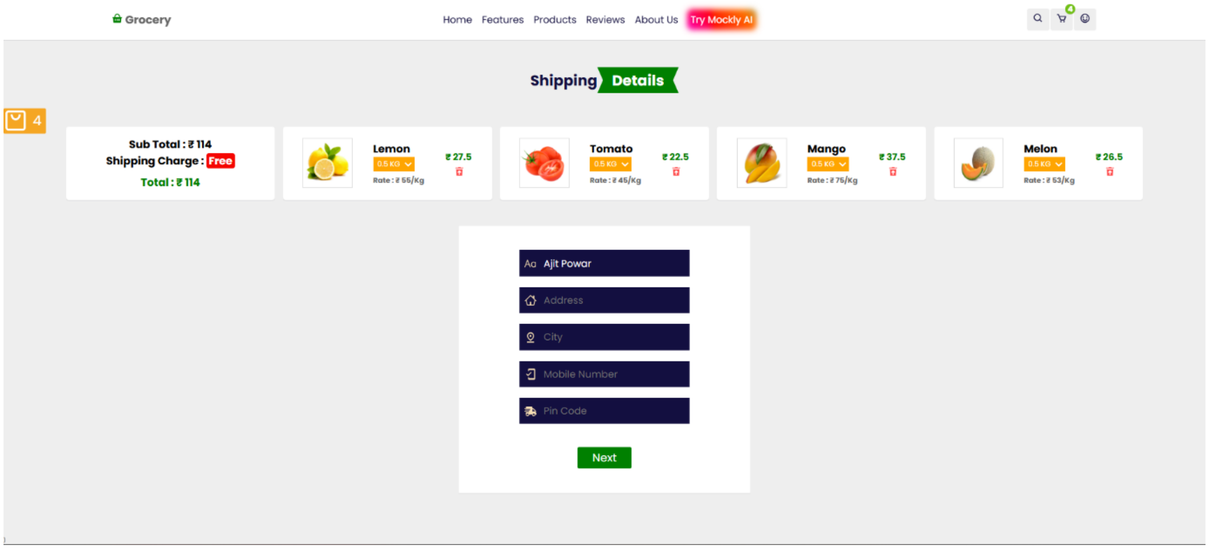
Cash On Delivery

We Offer Cash On Delivery...!!
You Can Pay When Your Products Is Delivered.

    ○ Product Catalog Service


Grocery
Home Features Products Reviews About Us Try Mockly AI
Fresh Products
Filter
Tomato ₹ 45/Kg 0.5 Kg
Lemon ₹ 55/Kg 0.5 Kg
Carrot ₹ 55/Kg OUT OF STOCK
Pineapple ₹ 70/Kg OUT OF STOCK
Strawberry ₹ 65/Kg OUT OF STOCK
Mango ₹ 75/Kg 0.5 Kg
Pappaya ₹ 65/Kg OUT OF STOCK
Melon ₹ 53/Kg 0.5 Kg
Banana ₹ 45/Kg OUT OF STOCK
Kiwi ₹ 100/Kg OUT OF STOCK
Coriander ₹ 10/Kg OUT OF STOCK
Gralic ₹ 49/Kg OUT OF STOCK
Leadyfinger ₹ 60/Kg OUT OF STOCK
Green Chilli ₹ 40/Kg 0.5 Kg
Pea ₹ 78/Kg OUT OF STOCK
Cauliflower ₹ 80/Kg OUT OF STOCK
Milk ₹ 25/Kg OUT OF STOCK
Eggplant ₹ 28/Kg OUT OF STOCK
0.5 KG IN CART
Groco
Fresh And Organic Delivery We Make Your Life Easy By Delivering
Contact Us
+91 7350403908
Quick Link
Home
Let's Chats
Type Message..!!

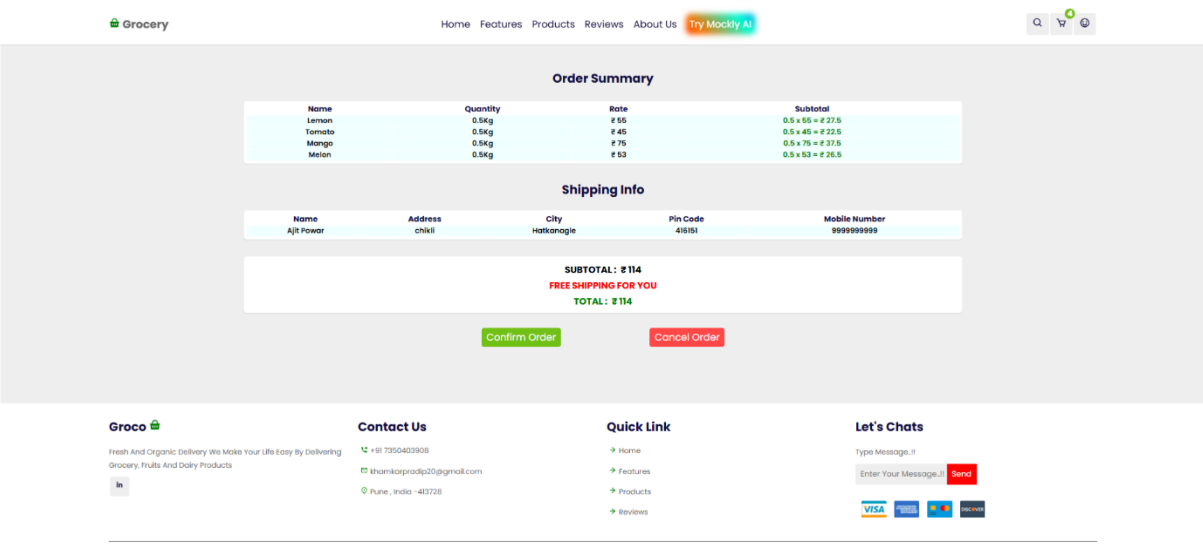
- ○. Shopping Cart Service


Grocery
Home Features Products Reviews About Us Try Mockly AI
Shipping Details
4
Lemon 0.5 KG Rate : ₹ 55/Kg ₹ 27.5
Tomato 0.5 KG Rate : ₹ 45/Kg ₹ 22.5
Mango 0.5 KG Rate : ₹ 75/Kg ₹ 37.5
Melon 0.5 KG Rate : ₹ 53/Kg ₹ 26.5
Address
City
Mobile Number
Pin Code
Next

Sub Total : ₹ 114
Shipping Charge : Free
Total : ₹ 114

Ajit Powar

- ○. Order Management Service



Order Summary

Name	Quantity	Rate	Subtotal
Lemon	0.5Kg	₴ 55	0.5 x 55 = ₴ 27.5
Tomato	0.5Kg	₴ 45	0.5 x 45 = ₴ 22.5
Mango	0.5Kg	₴ 75	0.5 x 75 = ₴ 37.5
Melon	0.5Kg	₴ 53	0.5 x 53 = ₴ 26.5

Shipping Info

Name	Address	City	Pin Code	Mobile Number
Ajit Powar	chikli	Hatkanagle	416151	9999999999

SUBTOTAL : ₴ 114

FREE SHIPPING FOR YOU

TOTAL : ₴ 114

Groco

Fresh And Organic Delivery We Make Your Life Easy By Delivering Grocery, Fruits And Dairy Products

Contact Us

+91 7350403908

khamkarpradip20@gmail.com

Pune , India -413728

Quick Link

→ Home

→ Features

→ Products

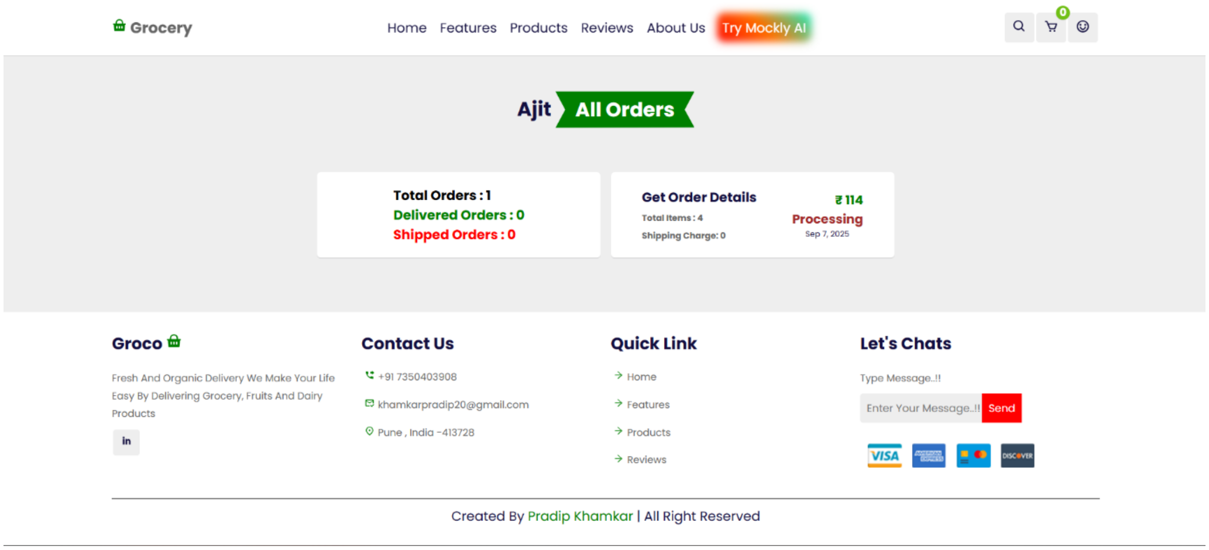
→ Reviews

Let's Chats

Type Message..!!

Enter Your Message..!!

○. Payment Service



Ajit All Orders

Total Orders : 1

Delivered Orders : 0

Shipped Orders : 0

Get Order Details ₴ 114

Total Items : 4

Shipping Charge: 0

Processing

Sep 7, 2025

Groco

Fresh And Organic Delivery We Make Your Life Easy By Delivering Grocery, Fruits And Dairy Products

Contact Us

+91 7350403908

khamkarpradip20@gmail.com

Pune , India -413728

Quick Link

→ Home

→ Features

→ Products

→ Reviews

Let's Chats

Type Message..!!

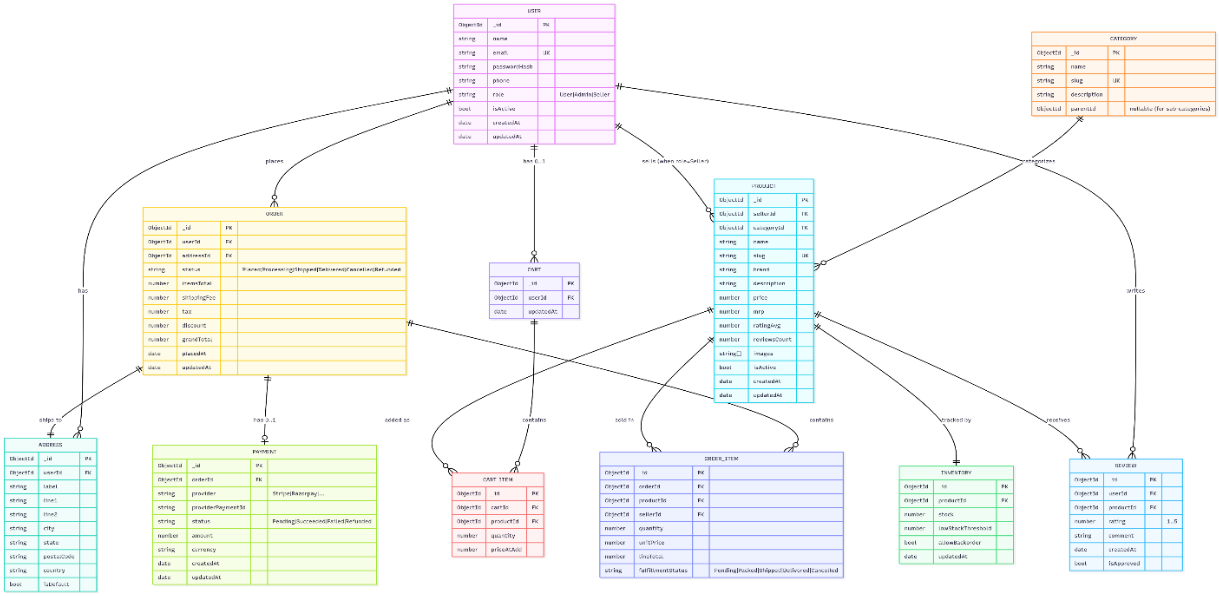
Enter Your Message..!!

Created By Pradip Khamkar | All Right Reserved

3. **Data Tier (MongoDB):** Responsible for data storage and retrieval.

4.2. Data Model (ER Diagram)

The Entity-Relationship Diagram below illustrates the key entities and their relationships, forming the blueprint of the database.



USER

ObjectId	_id	PK	
string	name		
string	email	UK	
string	passwordHash		
string	phone		
string	role		User|Admin|Seller
bool	isActive		
date	createdAt		
date	updatedAt		

CATEGORY

ObjectId	_id	PK	
string	name		
string	slug	UK	
string	description		
ObjectId	parentId		nullable (for sub-categories)

ORDER

ObjectId	_id	PK	
ObjectId	userId	FK	
ObjectId	addressId	FK	
string	status		Placed|Processing|Shipped|Delivered|Cancelled|Refunded
number	itemsTotal		
number	shippingFee		
number	tax		
number	discount		
number	grandTotal		
date	placedAt		
date	updatedAt		

CART

ObjectId	_id	PK
ObjectId	userId	FK
date	updatedAt	

PRODUCT

ObjectId	_id	PK
ObjectId	sellerId	FK
ObjectId	categoryId	FK
string	name	
string	slug	UK
string	brand	
string	description	
number	price	
number	mrp	
number	ratingAvg	
number	reviewsCount	
string[]	images	
bool	isActive	
date	createdAt	
date	updatedAt	

ADDRESS

ObjectId	_id	PK
ObjectId	userId	FK
string	label	
string	line1	
string	line2	
string	city	
string	state	
string	postalCode	
string	country	
bool	isDefault	

PAYMENT

ObjectId	_id	PK	
ObjectId	orderId	FK	
string	provider		Stripe|Razorpay|...
string	providerPaymentId		
string	status		Pending|Succeeded|Failed|Refunded
number	amount		
string	currency		
date	createdAt		
date	updatedAt		

CART_ITEM

ObjectId	_id	PK
ObjectId	cartId	FK
ObjectId	productId	FK
number	quantity	
number	priceAtAdd	

ORDER_ITEM

ObjectId	_id	PK	
ObjectId	orderId	FK	
ObjectId	productId	FK	
ObjectId	sellerId	FK	
number	quantity		
number	unitPrice		
number	lineTotal		
string	fulfillmentStatus		Pending|Packed|Shipped|Delivered|Cancelled

INVENTORY

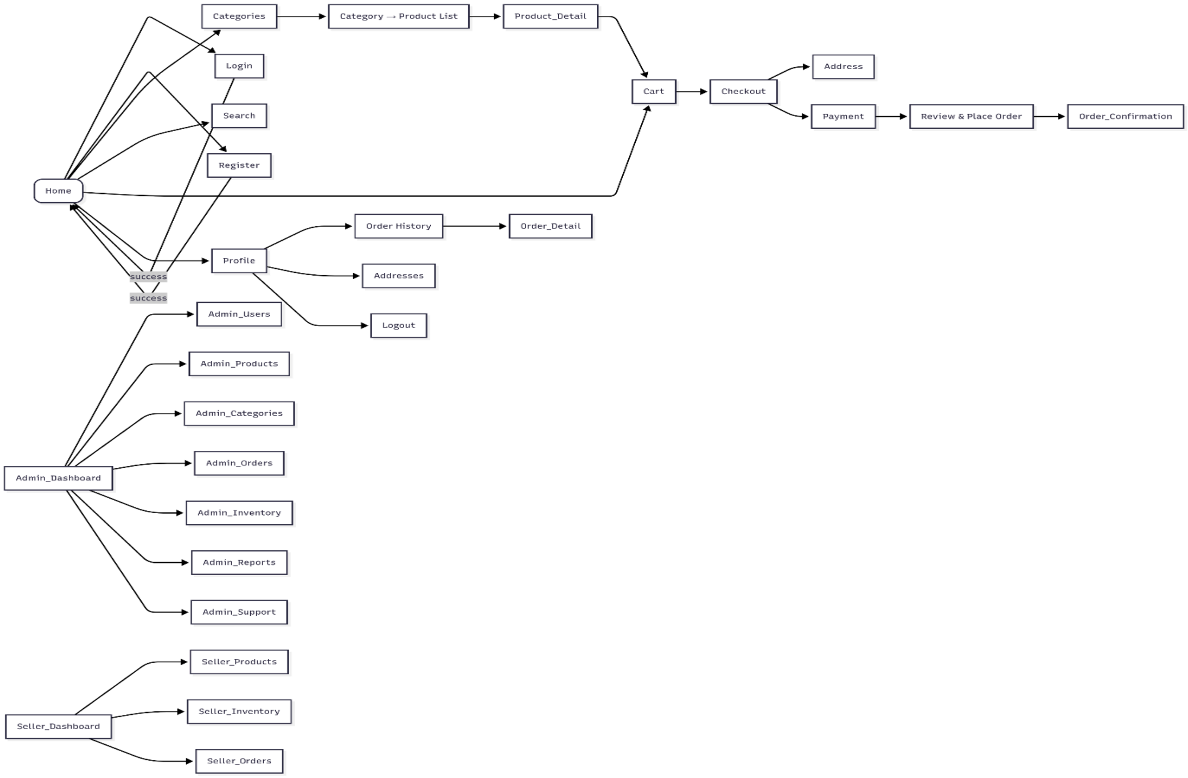
ObjectId	_id	PK
ObjectId	productId	FK
number	stock	
number	lowStockThreshold	
bool	allowBackorder	
date	updatedAt	

REVIEW

ObjectId	_id	PK	
ObjectId	userId	FK	
ObjectId	productId	FK	
number	rating		1..5
string	comment		
date	createdAt		
bool	isApproved		

4.3. Site Map & User Flow

The site map outlines the hierarchical structure of the application's pages and the flow a user follows to complete key tasks.



5. Development Setup & Deployment

5.1. Prerequisites

- **Node.js & npm:** JavaScript runtime and package manager. Download Here
- **MongoDB:** NoSQL database. Download Community Server or use MongoDB Atlas (cloud).
- **Angular CLI:** Command-line interface for Angular. Install via npm install -g @angular/cli.
- **Git:** Version control system. Download Here.
- **Code Editor:** Recommended: Visual Studio Code.

5.2. Cloning & Running the Project

1. **Clone the repository:**

```
git clone:

https://github.com/uborkar/Grocery_Web-App
```

2. **Install backend dependencies:**

```
npm install
```

3. **Configure Environment Variables:** Create a .env file in the root directory and set variables like MongoDB connection string (MONGODB_URI), JWT secret (JWT_SECRET), and port (PORT).

4. **Start the backend server:**

```
bash

npm run dev   # for development with auto-restart

# or

npm start     # for production
```

The backend API will run on http://localhost:5100 (or your specified PORT).

5. **Setup and Run the Frontend (Angular):**
   Note: The provided repository appears to be backend-centric. A separate Angular project is typically needed.

```
bash

# Navigate to your angular project directory (e.g., /client)

cd client

npm install

ng serve
```

The frontend will be available on http://localhost:4200.

Video Tutorial: A detailed setup guide is available here.

Project Repository: https://github.com/uborkar/Grocery_Web-App

6. Role-Based Access Control (RBAC)

Role	Permissions & Responsibilities
User/Customer	Register/Login, Browse catalog, Manage cart, Place orders, View own order history.
Admin	**All permissions of a Seller,** plus: Manage all users/roles, View platform-wide analytics and transactions, Manage system settings.
Seller	Add/Edit/Delete products and categories, Manage inventory stock, View and process orders for their products.

7. Project Development Flow

The project followed an iterative development lifecycle:

1. **Frontend Development (Angular):** Building UI components, pages, and services.
   - **UI/UX Design:** Wireframing and prototyping.
   - **Component Development:** Creating reusable Angular components (Product List, Cart, Checkout).
   - **Service Integration:** Writing services to connect to the backend API.
2. **Backend Development (Node.js):** Building the API and database logic.
   - **API Design:** Defining RESTful endpoints for all features.
   - **Database Modeling:** Creating Mongoose schemas and models.
   - **Authentication:** Implementing JWT-based secure login.
   - **Payment Integration:** Integrating a gateway like Stripe or Razorpay.
3. **Integration & Testing:** Connecting the frontend to the backend APIs, ensuring data flows correctly, and conducting end-to-end testing.

8. Conclusion

This Grocery Web Application SRS document outlines the vision, scope, and technical blueprint for a modern, scalable, and user-friendly e-commerce platform. By leveraging the MERN stack (MongoDB, Express.js, React.js, Node.js), the project is well-positioned to meet its functional and non-functional goals, delivering a valuable tool for consumers and administrators alike. Successful execution will depend on adherence to this plan, continuous testing, and iterative feedback incorporation.